A magyar nyelv néhány szófaji elemzőjének összevetése

Kuba András[1], Bakota Tibor[1], Hócza András[1], Oravecz Csaba[2]

[1] Szegedi Tudományegyetem, Informatikai Tanszékcsoport
MTA-SZTE Mesterséges Intelligencia Kutatócsoport
andkuba@inf.u-szeged.hu, bakotat@math.u-szeged.hu, hocza@inf.u-szeged.hu
[2] MTA Nyelvtudományi Intézet
oravecz@nytud.hu

Kulcsszavak: szófaji egyértelműsítés, szabály alapú módszerek, Hidden Markov modell

Absztrakt. A dolgozatban három különböző POS tagger (szófaji egyértelműsítő) összehasonlítására vállalkozunk. Az első egy Hidden Markov Model alapú bigram elemző (*VMM*), a második egy szabály alapú módszer, amely bizonytalansági osztályok felhasználásával szófaji egyértelműsítést végez (*RGLearn*). Mindkét elemző a Szegedi Tudományegyetem Informatikai Tanszékcsoportján készült. A harmadik egyértelműsítő a jól ismert *TnT* [1], amely már több nyelven bizonyította képességeit, és amely a VMM-el szemben a szövegben előforduló szóhármasokat vizsgálja. Kísérleteinket a körülbelül 1,2 millió szót tartalmazó, kézzel annotált *Szeged Korpuszon* [2] végeztük, amely különböző szövegtípusokat foglal magába. Vizsgálatunk tárgya a szófaji egyértelműsítés, vagyis a mondatban előforduló adott szóra a lehetséges kódok közül a mondat szemantikáját visszatükröző egyértelmű tag meghatározása. Azaz a tesztelés során az egyes szavak bizonytalansági osztálya ismert volt az elemzők előtt. Ez alól a TnT kivétel, mivel ez a módszer a tesztelés során a szóvégződések elemzése által következtet az ismeretlen szavak lehetséges nyelvtani kódjára. A tesztelés során az RGLearn algoritmus 96,16% pontosságával megelőzte a VMM elemzőt (95,98%) illetve a TnT-t (95,08%). A hibásan taggelt szavak listájának összehasonlítása során kiderült, hogy a két statisztikai módszer "hajlamosabb" ugyanazokon a helyeken hibázni. A kapott eredményeket felhasználva, vizsgálatokat végeztünk arra nézve is, hogy a fenti módszereket kombinálva milyen találati pontosság érhető el.

1. Bevezető

Természetes nyelvi szövegek szófaji címkézése (*taggelése*) az egyik legalapvetőbb számítógépes nyelvészeti feladat. A nemzetközileg publikált módszerek közül a magyar nyelvre azonban csak néhányat ültettek át. [3] A legelterjedtebb eljárások közé a statisztikai illetve a szabály alapú módszerek tartoznak. A Szegedi Tudományegyetem, Informatikai Tanszékécsoportján több szófaji egyértelműsítő is kifejlesztésre került, ezeket hasonlítottuk össze más ismert módszerekkel több szempont szerint.

Hangsúlyozni szeretnénk, hogy a programok kizárólag szófaji egyértelműsítést végeznek, így az egyes szavak lehetséges kódjait a bemenettel együtt megkapják. Ennek a jelentős egyszerűsítésnek a fő oka, hogy ezeket a módszereket egy programlánc *(ToolChain)* [4,5] részeként használjuk, melynek egy korábbi fázisában a *HuMor* [6] morfológiai elemző előállítja az egyes szavak bizonytalansági osztályait. Ezzel biztosítjuk, hogy az egyértelműsítés során a módszerek nem találkoznak olyan szóval, amelynek nem ismertek a lehetséges szófaji besorolásai.

A legtöbb nemzetközileg ismert elemző, mint például a *TnT* is, saját beépített morfológiai elemzővel rendelkezik, mely általában a tanulás során előforduló szóvégződések, illetve a prefixek segítségével naiv következtetéseket tud levonni a szavak lehetséges tagjeit illetően. Az elemzők összehasonlítása során nem volt módunk arra, hogy a *TnT* előtt ismerté tegyük a szavak valódi bizonytalansági osztályát, így ez a módszer a többihez képest hátránnyal indult.

Kísérleteinket a körülbelül 1,2 millió szót tartalmazó, kézzel annotált Szeged Korpuszon végeztük, amely különböző szövegtípusokat foglal magába. A korpusz nagyon részletes MSD kódolást használ, az egyes tagek jelölésére, így az előforduló különböző tagek száma meghaladja az 1400-at.

A következő fejezetben a *VMM, TnT* illetve az *RGLearn* taggerek jellegzetességeit ismertetjük.

2. A VMM tagger

A VMM valójában egy Hidden Markov Modellt megvalósító algoritmus. [7] A tanulás során a Modell paraméterei közvetlenül számíthatók, mivel a tréning alapjául szolgáló korpuszban a szavak helyes szófaji kódjai be vannak jelölve. Más szavakkal a modellben az állapotátmenetek ismertek a tanulás ideje alatt, éppen ezért *Visible Markov Model* néven is emlegetik. A tréning során nem csak a modell paraméterei kerülnek kiszámításra, hanem statisztikák készülnek az egyes bizonytalansági osztályokra is: melyik osztály hányszor fordult elő, melyik volt a leggyakrabban kiválasztott szófaji kód, és az hányszor bizonyult helyesnek. Ha a tesztelés során olyan szót vizsgálunk, amely nem fordult elő a tanítás során, akkor az illető szó bizonytalansági osztályában előforduló kódokhoz kezdeti eloszlást rendelünk, amely megfelel a korpuszból előzetesen kigyűjtött adatoknak. Az egyértelműsítő módszer a számára ismert szavakhoz azt a kód-eloszlást rendeli, amely az adott szóra a tanulás során kialakult. Az egyértelműsítés a Viterbi algoritmus [7] segítségével történik.

A többféle tesztelés során három szinten mértük az egyértelműsítő módszer találati pontosságát:

1. szint: a módszer által eredményül adott, illetve a korpuszban kézzel meghatározott kódok első karakterének egyezése – szófaj meghatározás,
2. szint: összevont kódrendeszer – az egymástól lényegesen nem különböző szófaji címkéket csoportokba rendeztük, és a csoporton belüli címkéket azonosnak vettük, azaz itt nem követeltünk meg teljes egyezést
3. szint: az *MSD* kódok teljes egyezése

A találati pontosságot nem csak az összes szó arányában, hanem a többértelmű szavak tekintetében is megvizsgáltuk. Ugyanis hibás döntést csak a többértelmű szavakon hozhat az algoritmus (ahol egy szónak több különböző szófaji besorolása is lehetséges).

Vizsgáltuk többek közt a tagger átlagos viselkedését (*90-10 cross fold validation*), melynek az eredményeit az alábbi táblázatban foglaltuk össze:

Elfogadási szint	Min	Max	Eltérés	Átlag
1. szint	97.11%	97.67%	0.56%	97.48%
2. szint	95.74%	96.38%	0.64%	96.16%
3. szint összes szóra vetítve	95.52%	96.17%	0.65%	95.93%
3. szint a többértelmű szavakra vetítve	90.26%	91.69%	1.43%	91.20%

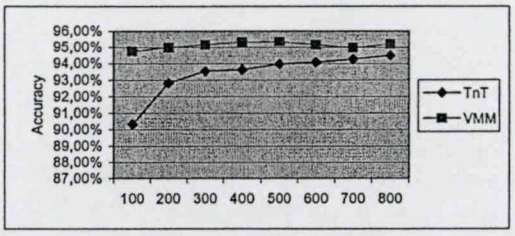
1. Táblázat: 90-10 cross-fold validáció eredményei az egyes szintekre lebontva

3. TnT (Trigrams'n'Tags)

A *TnT* szintén statisztikai elven működő szófaji egyértelműsítő program, amelyet *Thorsten Brants (Saarland University)* fejlesztett ki a 90-es évek elején. [1] Előnye, hogy tetszőleges nyelvre alkalmazható, nagy találati pontosságot képes elérni, és gyors. Hátránya azonban, hogy az egyes szavak bizonytalansági osztálya nem adható meg közvetlenül, hanem saját beépített morfológiai elemzőjével igyekszik meghatározni a lehetséges szófaji címkéket. Ha a szófaji besorolás nem túl részletes, azaz nincs sok

lehetséges címke, akkor nagyon jó eredményeket produkál (96-97% per word), azonban a mi vizsgálatainkban 1-1,5%-kal lemarad a többi taggertől a találati pontosságot tekintve.

A tesztelések során vizsgáltuk a TnT találati pontosságát, illetve, hogy az hogyan változik a tréninghalmaz növelésével. Az eredményeket összehasonlítottuk a VMM – nél kapott adatokkal:



1. Ábra: A találati pontosság alakulása a tréning méretének növelésével a TnT, illetve a VMM elemzők esetében. A tréning mérete a korpusz fájlok számával van megadva.

4. Szabály alapú taggerek

A szabály alapú módszereknek számos előnyük van a statisztikai taggerekhez képest:

- A szabályok könnyen áttekinthetők, értelmezhetők,
- Könnyen kiegészíthetők szakértői tudás beépítésével, amelyek megnyilvánulhatnak szakértők által adott szabályokban, kezdeti hipotézisben, vagy a meglévő szabályok finomításában
- A támogatás gyors és egyszerűen megvalósítható.

4.1 Az RGLearn szabályrendszer tanuló algoritmus

Az RGLearn egy saját fejlesztésű algoritmus, amely egy kezdeti szabályrendszerből kiindulva azt úgy próbálja általánosítani, hogy a tréning példákat a lehető legkevesebb számú, minél általánosabb szabállyal lefedje, úgy hogy a szabályok hibája (amikor rossz döntést hoznak) egy adott küszöbérték alatt maradjon. A kezdeti szabályrendszer lehet más tanuló módszerek vagy nyelvész szakértők által előállított szabályrendszer is. Az általunk használt kezdeti szabályrendszer a nem alapértelmezett választást tartalmazó tréning példákat tartalmazta. Alapértelmezett választás az a szófaji kód, amely a leggyakrabban előfordul az adott szóra nézve.

```
RULE_SET = non default cases from EXAMPLE_SET
while change RULE_SET do
{
     foreach RULE of RULE_SET do unification RULE
     foreach RULE of RULE_SET do generalization RULE
     foreach RULE of RULE_SET do delete rules covered by RULE
}
```

Unification RULE:

1.) Megkeresi azt a szabályt ami a RULE szabályhoz legjobban hasonlít (az attribútumok értékei (szavak, nyelvtani kódok) a legtöbb karakter pozíción egyeznek az értékek elejétől).
2.) A két szabályt összevonja (a különböző részeket elhagyja)
3.) A két szabály helyett bevezeti az összevont szabályt ha annak pontossága nagyobb egy előre megadott küszöbértéknél.

Generalization RULE:

1.) Készít egy új szabályt RULE szabályból úgy, hogy a környezet szélétől befelé haladva egy attribútum értékét általánosabb reguláris kifejezésre cseréli vagy elhagyja.

3.) A RULE szabály helyett bevezeti az általánosabb szabályt ha annak pontossága nagyobb egy előre megadott küszöbértéknél.

4.2. A szabály alapú szófaji egyértelműsítő működése

A szófaji egyértelműsítő által használt szabályrendszer többféle szempont szerint rendezett. Ezáltal gyors (bináris) keresés valósítható meg a döntéshez szükséges rész-szabályhalmaz kiválasztásához. A szabályok kiválasztásakor a sok példát lefedő, minél pontosabb szabályokat próbáljuk alkalmazni először. Az egyértelműsítés mondatonként történik, esetenként több menetben amíg a tagger talál új egyértelműsíthető szót. Ha egy adott többértelmű szóra nincs szabály az az alapértelmezett (leggyakoribb) nyelvtani kódot kapja meg a bizonytalansági osztály választási lehetőségei közül. A szófaji egyértelműsítés algoritmusa a következő:

```
while change do
{
      foreach tag of sentence do
      if tag not decided then
      foreach rule of ruleset do
      if rule covers tag then decide code of tag by using rule
}
foreach tag of sentence do
if tag not decided then set default code of tag
```

4.3. A C4.5 döntésifa tanuló algoritmus

A C4.5 algoritmus az ID3-algoritmus egy továbbfejlesztett változata. J. R. Quinlan nevéhez fűződik. [8] A C4.5 egy döntési fát állít elő, melyben a csomópontok egy-egy attribútumra vonatkozó kérdések, a levelek pedig a döntések. A C4.5 úgy próbálja előállítani a döntési fát, hogy minél kevesebb kérdéssel el lehessen jutni a döntéshez, ezért azokat az attribútumokat választja ki csomópontoknak, melyeknek legnagyobb az információs nyeresége. A döntési fa pedig átkonvertálható szabályokká.

5. Eredmények

Az alábbi táblázat a 4 tárgyalt egyértelműsítő módszer, és referenciaként a jól ismert C4.5 módszer által elért eredményeket mutatja egy konkrét teszt esetén.

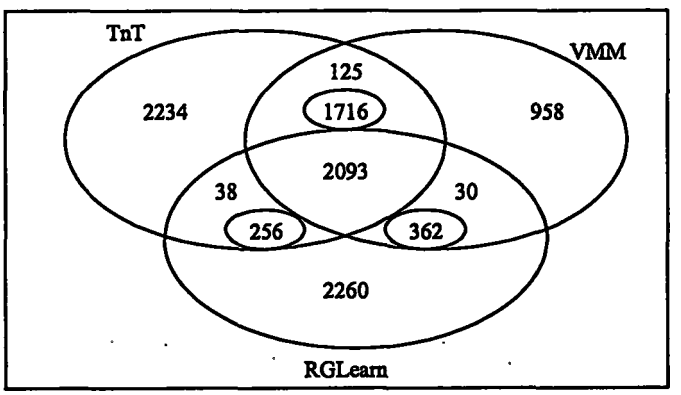
	VMM	TnT	C4.5	RGLearn
Tréning idő	10 perc	30 mp	~6 óra	~24 óra
Teszt idő	11 perc	30 mp	3 perc	2 perc
Tréning fájlok száma	823	823	823	823
Teszt fájlok száma	91	91	91	91
Szavak száma	131345	131345	131345	131345
Többértelmű szavak száma	60758	60758	60758	60758
Hibásan jelölt szavak száma	5284	6462	6646	5039
Hibásan jelölt szavak (1. szint)	3435	-	4338	3043
Pontosság				
3. szint az összes szón	95,98%	95,08%	94,94%	96,16%
3. szint a többértelmű szavakon	91,30%	89,36%	89,06%	91,71%
1. szint az összes szón	97,38%	-	96,70%	97,68%
1. szint a többértelmű szavakon	94,35%	-	92,86%	94,99%

2. Táblázat: A három egyértelműsítő módszer által ugyanazon a tréning és teszt adaton produkált eredmények

A módszerek közül az RGLearn érte el a legjobb eredményt. Figyelemre méltó, hogy az alapvetően más megközelítéssel dolgozó módszerek mennyire hasonló eredményeket produkálnak. Ha a teljes egyezés helyett megelégszünk az első szintű egyezéssel, akkor minden módszer eredménye nagyjából 1,5 százalékot javul. Nagy különbség van azonban az egyes módszerek tanulási és futási időigénye között. Erre feltétlenül figyelemmel kell lenni, ha a módszereket alkalmazni kívánjuk.

5.1. Kombinált módszerek

A következőkben csak a három legjobb eredményt produkáló egyértelműsítő algoritmusra koncentrálunk. Az algoritmusok nem csak pontosságban, hanem az elkövetett hibák típusában is különböznek. Az egyes módszerek esetében érdemes összevetni a helytelenül taggelt szavak listáját. Az alábbi ábrán látható a hibásan megjelölt szavak száma a fenti teszt esetében:



2. Ábra: Az egyes módszerek által hibásan megjelölt szavak eloszlása. Az egyes halmazok ábrázolják az adott módszer által hibásan jelölt szavakat. A metszetekben (ahol egy módszer helyes, kettő pedig helytelen eredményt ad) külön csoportosítottuk azokat a szavakat, amelyekre a két hibázó módszer ugyanazt a (hibás) eredményt adta.

A kérdés, hogy lehet-e a legpontosabb tagger algoritmusnál is jobb eredményt elérni egy kombinált módszer alkalmazásával, amely figyelembe veszi a kevésbé pontos algoritmusok eredményeit is?

Ahol a 3 halmaz metszi egymást, azaz mindhárom tagger hibás eredményt ad, ott egy kombinált módszer sem segíthet, de azon szavak esetében, ahol legalább az egyik tagger jól dönt, van esély a helyes megoldás megtalálására. A kialakuló kombinált módszer hatékonysága nyilván a döntési stratégián múlik. A döntési stratégia mondja meg, hogy ha egy szó esetén mindhárom módszer által adott eredményt ismerjük, ezek közül melyiket válasszuk. Mi olyan döntési stratégiákat vizsgáltunk, amelyek függetlenek az egyes módszerek által választott tagektől.

Ha a 3 módszer 3 különböző választ ad, akkor valamelyik módszert ki kell tűntetnünk, és a preferált módszer által szolgáltatott eredményt fogadjuk el véglegesnek. Az ábráról leolvasható, hogy ilyen esetben az RGLearn algoritmust érdemes preferálni, mivel ez 125 esetben ad helyes választ a másik két módszer 30, illetve 38 találatával szemben.

Ha a módszerek által szolgáltatott tagek közül kettő megegyezik, de a harmadik eltér ettől, akkor alapvetően két dolgot tehetünk: vagy az egyező taget választjuk, vagy a különbözőt. Az ábráról leolvasható, hogy (nem meglepő módon) ilyenkor azt a taget érdemes választani, amelyiket két módszer egyformán eredményül adta.

Ha a módszerek által adott mindhárom tag megegyezik, nyilván nincs módunkban mást tenni, mint ezt a taget eredményül adni.

A fentiek értelmében tehát az elérhető legjobb algoritmus a korábban látott teszt esetén az, hogy a három módszert megszavaztatjuk: mindhárom módszer eredményének ismeretében azt a taget választjuk. Amelyre legalább két módszer szavazott, ha pedig nincs ilyen, akkor az RGLearn módszer által adott eredményt választjuk. Ezzel a stratégiával az elérhető pontosság az összes szóra vetítve 96,58%-nak, a többértelmű szavakon 92,6%-nak adódik. Azaz a pontosság a többértelmű szavakon a legjobb módszerhez képest is kb. 1 százalékot javult.

6. Összegzés

A Szeged Korpusz jó alapot ad ahhoz, hogy a különböző szófaji egyértelműsítő módszereket összehasonlíthassuk. A négy vizsgált módszerből (a C4.5 algoritmust is ideértve) kettő szabályalapú, kettő pedig statisztikai volt; mindkét csoportból volt egy standard, jól bevált eszköz (C4.5 és TnT) valamint egy általunk megvalósított módszer (VMM és RGLearn). A módszerek nagyjából hasonló eredményt produkálnak, a legjobb eredményt az RGLearn algoritmus adta.

Vizsgáltuk azt is, hogy nagyobb szövegen javulna-e a módszerek pontossága. A tréning adatok további növelésével lényegesen jobb eredményt már nem várhatunk.

A pontosság tovább növelhető viszont, ha a rendelkezésre álló módszereket párhuzamosan felhasználjuk valamilyen kombinált módszerben. Egy ilyen algoritmussal az egyértelműsítés pontosságát kb. 1 százalékkal javítani tudtuk a legjobb algoritmushoz képest.

A kutatás eredménye tehát egy viszonylag jó pontosságú szófaji egyértelműsítő program lett, amely bármilyen magyar nyelvű korpusz annotálására használható. Ez a jövőben egy olyan rendszer modulja lesz, amely magyar nyelvű természetes szöveget dolgoz fel információ-kinyerés céljából. [4,5]

Az automatikus módszerek a kézzel annotált korpuszok hibajavításában is segíthetnek. Jelenleg a Szeged Korpusz munkálataiban hibajavításra használjuk az itt ismertetett módszereket. Azokat a szavakat, ahol az automatikus módszer hibázik, nyelvész szakértők átnézik, és amennyiben szükséges, javítják. A cikk írásáig körülbelül 8000 hibásan annotált szóra derült így fény. Ez az összes szó kb. 0,7 százaléka.

Irodalom

1. Brants, T.: *TnT – A Statistical Part-of-Speech Tagger*, Saarland University, Computational Linguistics (2000)
2. Alexin, Z., Csirik, J., Gyimóthy, T., Bibok, K., Hatvani, Cs., Prószéky, G., Tihanyi, L. (2003) *Manually Annotated Hungarian Corpus*, in Proceedings of the Research Note Sessions of the 10th Conference of the European Chapter of the Association for Computational Linguistics EACL03, Budapest, Hungary, pp. 53–56.
3. Horváth, T., Alexin, Z, Gyimóthy, T. and Wrobel, S. *Application of Different Learning Methods to Hungarian Part-of-speech Tagging*, in Proceedings of 9th International Workshop on Inductive Logic Programming (ILP99) Bled, Slovenia, in the LNAI series Vol 1634 p. 128–139, Springer Verlag (1999)
4. Hócza, A., Alexin, Z., Csendes, D., Csirik, J., Gyimóthy, T.: *Application of ILP methods in different natural language processing phases for information extraction from Hungarian texts* in Proc. of the Kalmár Workshop on Logic and Computer Science, Szeged, Hungary, 1-2 October, pp. 107-116 (2003)
5. Freitag D. *Machine Learning for Information Extraction in Informal Domains*, Machine Learning, 39, 169–202. (2000)
6. Prószéky, Gábor: *Humor: a Morphological System for Corpus Analysis*, Language Resources for Language *Technology* (Proceedings of the First European TELRI Seminar), 149-158. Tihany, Hungary (1995)
7. Manning, C. D., Schütze, H.: *Foundations of Statistical Natural Language Processing*, Chapter 9. MIT Press (1999)
8. Quinlan, J. R. *C 4.5: Programs for Machine Learning*, Morgan Kaufmann Publisher. (1993)